



A MODIFIED GAIN SCHEDULING CONTROLLER BY CONSIDERING THE SPARSENESS PROPERTY OF UAV QUADROTORS

**M. Qodar Abdurrohman^{a,*}, Reka Inov^a, Ahmad Ataka^a, Hilton Tnunay^a,
Ardhimas Wimbo^a, Iswanto^a, Adha Cahyadi^a, Yoshio Yamamoto^b**

^aDepartment of Electrical Engineering and Information Technology, Gadjah Mada University
Jalan Grafika 2, Yogyakarta 55281

^bDepartment of Precision Engineering, School of Engineering, Tokai University
1117 Hiratsuka-Shi, Kanagawa-Ken, Japan

Received 12 November 2014; received in revised form 17 March 2015; accepted 17 March 2015
Published online 30 July 2015

Abstract

This work presented the gain scheduling based LQR for Quadrotor systems. From the original nonlinear model, the system is always controllable and observable in various equilibrium points. Moreover, the linearized systems have a unique property that is known as sparse system. Hence, in order to implement the most efficient state feedback controller, post-filter and pre-filter were introduced to transform the state coordinate to decrease coupling between states. Finally, the gain scheduling systems using these facts was proposed. The system behavior was tested using the proposed controller. The numerical studies showed the effectiveness of the controller to achieve desired altitude, attitude, and its ability during the disturbance.

Keywords: quadcopters; sparse system; linearization; gain scheduling; pole-placement.

I. INTRODUCTION

Quadcopter is one of Unmanned Aerial Vehicles that become popular and having much attention recently, especially from the researchers and hobbyist in aeromodelling. Several factors that contribute to its popularity are its reliability in maneuvering, its ability to be flown indoors, and easier to model and control [1], [2].

One of the most important problem on the quadcopter stems comes from the fact that quadcopter is essentially not a stable system, both in stabilization and trajectory following. Therefore, special considerations are needed in designing the control system for stabilizing or maneuvering. Existing control theories in controlling quadcopter are widely varied. The most commonly used is the conventional PID control [3], mainly due to its simple structure that is easy enough to be designed and implemented in varied systems, including quadcopter [1].

The drawback of PID controller is the gain that set for optimum in some specific conditions. In order to get the better results, the controller

has to be adaptive so that it can adjust the controller gain to adapt to the position and attitude change of the quadrotor. Many people have tried to design this adaptive control such as Gaikwad [4] with auto-tuning PID Loop Shaping and Liu [5] who design self-adaptive PID based on the least-square method. Another approach is proposed to control a quadcopter using PD controller equipped with active force control to reject uncertainty disturbance by estimating disturbance torque value [6].

One of the challenges in designing controller for quadrotor is that the system is non-linear, a very common to linearize first. The basic limitation of the controller design via standard linearization is the fact that the control is guaranteed to work only in the neighborhood of a single equilibrium point. Gain scheduling is a technique to design a controller of non-linear system by linearization the system at several equilibrium points, designing the controller at each point, and implementing the family of linear controllers as a single controller with varying gain or parameter [7].

This paper also present the linearization of the simplified model of quadrotor based on [1] using

* Corresponding Author. Tel: +6281804092541
E-mail: m.qodar.a@mail.ugm.ac.id

the gain scheduling linearization at some equilibrium points. Then, the controllability and observability of the system for various equilibrium points is proposed. After that, using the linearized state equations, the state feedback controllers is obtained and applied to control the altitude and attitude of the quadrotor non-linear model. For implementation purpose, a controller that focused on improvement of pole placement method is proposed by restructuring the state variable of linearized quadcopter dynamics to its Jordan form to emphasize the sparseness of the dynamics.

II. SYSTEM OVERVIEW

A. Notation Explanation

This section consists of explanation on the derivation of state-space equations of quadcopter system. The linearization is performed to formulate the transfer function of the quadcopter plant. The model is shown in Figure 1. 12 states are used for this state-space model. The position in world frame is denoted as $\epsilon_1 = [x \ y \ z]$, while the roll, pitch, yaw angles denoted as $\epsilon_2 = [\phi \ \theta \ \psi]$. The velocity due to x-axis, y-axis, z-axis denoted as $\epsilon_3 = [\dot{x} \ \dot{y} \ \dot{z}]$, and the angular velocity due to x-axis, y-axis, z-axis denoted as $\epsilon_4 = [\dot{p} \ \dot{q} \ \dot{r}]$. The state variable and its system input is set as $x = [\epsilon_1^T \ \epsilon_2^T \ \epsilon_3^T \ \epsilon_4^T]^T$, and $u = [\omega_1^2 \ \omega_2^2 \ \omega_3^2 \ \omega_4^2]$. Hence it was set that $x = [x_1, \dots, x_{12}]^T$ and $u = [u_1, \dots, u_4]^T$.

B. Translational and Rotational Analysis

Based on Newton's second law of translational motion, this equation is obtained:

$$F = m\ddot{v} + (\omega \times mv) \quad (1)$$

where $\omega = \epsilon_4$ and $v = \epsilon_3$. From Figure 1, the forces which is worked on the quadcopter is obtained as:



Figure 1. Quadcopter axis

$$F = F_g - F_{\text{thrust}} \quad (2)$$

$$F = [0 \ 0 \ mg]^T + {}^wR_B [0 \ 0 \ T]^T \quad (3)$$

Therefore, equation 3 can be expressed as:

$$\dot{v} = \left[\frac{1}{m} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} - {}^wR_B \begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix} \right] - \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} \times \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} \quad (4)$$

where m is the mass of quadcopter, T is vertical thrust of quadcopter against gravity and wR_B is the rotation matrix from body-frame to world-frame or inertial-frame, where:

$${}^wR_B = \begin{bmatrix} c_\phi c_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi & c_\phi s_\theta c_\psi + s_\phi s_\psi \\ c_\phi s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi \\ s_\theta & s_\phi c_\theta & c_\phi c_\theta \end{bmatrix} \quad (5)$$

$$\dot{v} = \left[\frac{1}{m} \begin{bmatrix} -T(c_\phi s_\theta c_\psi + s_\phi s_\psi) \\ -T(c_\phi s_\theta s_\psi - s_\phi c_\psi) \\ mg - Tc_\phi c_\theta \end{bmatrix} - {}^wR_B \begin{bmatrix} \dot{q}\dot{z} - \dot{r}\dot{y} \\ \dot{r}\dot{x} - \dot{p}\dot{z} \\ \dot{p}\dot{y} - \dot{q}\dot{x} \end{bmatrix} \right] \quad (6)$$

Assumed that $\dot{p}, \dot{q}, \dot{r}, \dot{x}, \dot{y}, \dot{z}$ equal to zero, then:

$$\ddot{x} = -\frac{1}{m}T(c_\phi s_\theta c_\psi + s_\phi s_\psi) \quad (7)$$

$$\ddot{y} = -\frac{1}{m}T(c_\phi s_\theta s_\psi - s_\phi c_\psi) \quad (8)$$

$$\ddot{z} = g - \frac{1}{m}Tc_\phi c_\theta \quad (9)$$

Using rigid body rotational law, Γ is:

$$\Gamma = I\dot{\omega} + (\dot{\omega} \times I\omega) \quad (10)$$

where, I is moment of inertia of quadcopter as:

$$I = \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} \quad (11)$$

Then, equation 10 can be written as:

$$I\dot{\omega} = \begin{bmatrix} \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} - \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} \times \begin{bmatrix} I_x \dot{p} \\ I_y \dot{q} \\ I_z \dot{r} \end{bmatrix} \quad (12)$$

and

$$\tau_x = db(\omega_4^2 - \omega_2^2) \quad (13)$$

$$\tau_y = db(\omega_1^2 - \omega_3^2) \quad (14)$$

$$\tau_z = k(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \quad (15)$$

$$T = b(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) \quad (16)$$

Thus, these equations for angular acceleration of quadcopter is:

$$\ddot{p} = \frac{db}{I_x}(\omega_4^2 - \omega_2^2) - \frac{I_z - I_y}{I_x} \dot{q}\dot{r} \quad (17)$$

$$\ddot{q} = \frac{db}{I_y}(\omega_1^2 - \omega_3^2) - \frac{I_x - I_z}{I_y} \dot{p}\dot{r} \quad (18)$$

$$\ddot{r} = \frac{k}{I_z}(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) - \frac{I_y - I_x}{I_z} \dot{p}\dot{q} \quad (19)$$

The roll, pitch and yaw (RPY) rates which is a function of angular velocity were derived using inverted Jacobian matrix, denoted as:

$$W^{-1} = \frac{1}{c_\theta} \begin{bmatrix} c_\theta & s_\phi s_\theta & c_\phi s_\theta \\ 0 & c_\phi c_\theta & -s_\phi c_\theta \\ 0 & s_\phi & c_\phi \end{bmatrix} \quad (20)$$

and the relation between RPY rates and angular velocity is expressed by matrix:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = W^{-1} \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} \quad (21)$$

So, the roll, pitch, yaw rates:

$$\dot{\phi} = \dot{p} + s_\phi t_\theta \dot{q} + c_\phi t_\theta \dot{r} = 0 \quad (22)$$

$$\dot{\theta} = c_\phi \dot{q} + s_\phi \dot{r} = 0 \quad (23)$$

$$\dot{\psi} = \frac{s_\phi}{c_\theta} \dot{q} + \frac{c_\phi}{c_\theta} \dot{r} = 0 \quad (24)$$

C. Linearization

The non-linear model of quadrotor will be linearized at the equilibrium points to make the system more amenable. In order to do the linearization, it is required to find the equilibrium point of the system, hence $0 = f(\chi(\alpha, \beta, \gamma, \delta))$. It is trivial that part of equilibrium points are as follow:

$$\dot{x}_1 = \dot{x} = \dot{x}_7 = 0 \quad (25)$$

$$\dot{x}_2 = \dot{x} = \dot{x}_8 = 0 \quad (26)$$

$$\dot{x}_3 = \dot{x} = \dot{x}_9 = 0 \quad (27)$$

$$\dot{x}_4 = \dot{\phi} = x_{10} + s_{x_4} t_{x_5} x_{11} + c_{x_4} t_{x_5} x_{12} = 0 \quad (28)$$

$$\dot{x}_5 = \dot{\theta} = c_{x_4} x_{11} + s_{x_4} x_{12} = 0 \quad (29)$$

$$\dot{x}_6 = \dot{\psi} = \frac{s_{x_4}}{c_{x_5}} x_{11} + \frac{c_{x_4}}{c_{x_5}} x_{12} = 0 \quad (30)$$

$$\dot{x}_7 = \ddot{x} = -\frac{1}{m} T(c_{x_4} s_{x_5} c_{x_6} + s_{x_4} s_{x_6}) = 0 \quad (31)$$

$$\dot{x}_8 = \ddot{y} = -\frac{1}{m} T(c_{x_4} s_{x_5} s_{x_6} - s_{x_4} c_{x_6}) = 0 \quad (32)$$

$$\dot{x}_9 = \ddot{z} = g - \frac{1}{m} T(c_{x_4} c_{x_5}) = 0 \quad (33)$$

$$\dot{x}_{10} = \ddot{p} = \frac{db}{I_x} (u_4 - u_2) - \frac{I_z - I_y}{I_x} x_{11} x_{12} = 0 \quad (34)$$

$$\dot{x}_{11} = \ddot{q} = \frac{db}{I_y} (u_1 - u_3) - \frac{I_x - I_z}{I_y} x_{10} x_{12} \quad (35)$$

$$\dot{x}_{12} = \ddot{r} = \frac{k}{I_z} (u_1 - u_2 + u_3 - u_4) - \frac{I_y - I_x}{I_z} x_{10} x_{11} \quad (36)$$

From equation (31) and (32), it was obtained:

$$s^2 x_5 + t^2 x_4 = 0 \quad (37)$$

The only solution for this equation is $x_4 = 0$ and $x_5 = 0$. By combining the results with equation (28), (29), and (30) following state can

be obtained as $x_{10} = 0$, $x_{11} = 0$ and $x_{12} = 0$. Using assumption that the equilibrium point is located at certain positions in Cartesian coordinate (x , y , z) and at some yaw angle positions is defined by $x = \alpha$, $y = \beta$, $z = \gamma$, and $\psi = \delta$. So, the complete list of the state variables value in this equilibrium, $X(\alpha, \beta, \gamma, \delta)$, can be written as $x_1 = \alpha$, $x_2 = \beta$, $x_3 = \gamma$, $x_4 = 0$, $x_5 = 0$, $x_6 = \delta$, $x_7 = 0$, $x_8 = 0$, $x_9 = 0$, $x_{10} = 0$, $x_{11} = 0$, and $x_{12} = 0$.

Remark 1. It can be noted that in the equilibrium, unless the arbitrary position and the yaw angle, all of the state are zeros.

The state equation for the linearized state space model is represented by:

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du. \end{aligned} \quad (38)$$

where the matrix A and B can be found by using following equation.

$$A = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} |X(\alpha, \beta, \gamma, \delta) & \cdots & \frac{\partial f_1}{\partial x_{12}} |X(\alpha, \beta, \gamma, \delta) \\ \vdots & \ddots & \vdots \\ \frac{\partial f_{12}}{\partial x_1} |X(\alpha, \beta, \gamma, \delta) & \cdots & \frac{\partial f_{12}}{\partial x_{12}} |X(\alpha, \beta, \gamma, \delta) \end{bmatrix} \quad (39)$$

$$B = \begin{bmatrix} \frac{\partial f_1}{\partial u_1} |X(\alpha, \beta, \gamma, \delta) & \cdots & \frac{\partial f_1}{\partial u_{12}} |X(\alpha, \beta, \gamma, \delta) \\ \vdots & \ddots & \vdots \\ \frac{\partial f_{12}}{\partial u_1} |X(\alpha, \beta, \gamma, \delta) & \cdots & \frac{\partial f_{12}}{\partial u_{12}} |X(\alpha, \beta, \gamma, \delta) \end{bmatrix} \quad (40)$$

By careful calculation it was found that:

$$A_{(12 \times 12)} = \begin{bmatrix} O_{(6 \times 6)} & I_{(6 \times 6)} \\ O_{(2 \times 2)} & N_{(2 \times 2)} & O_{(2 \times 1)} & I_{(6 \times 6)} \\ O_{(4 \times 6)} \end{bmatrix} \quad (41)$$

and

$$B_{(12 \times 4)} = \begin{bmatrix} O_{(8 \times 4)} \\ M_{(4 \times 4)} \end{bmatrix} \quad (42)$$

where O is zero matrix and I is identity matrix. While N and M can be defined as:

$$N_{(2 \times 2)} = \begin{bmatrix} -gs_\delta & -gc_\delta \\ gc_\delta & gs_\delta \end{bmatrix} \quad (43)$$

$$M_{(2 \times 2)} = \begin{bmatrix} -\frac{b}{m} & -\frac{b}{m} & -\frac{b}{m} & -\frac{b}{m} \\ 0 & -\frac{db}{I_x} & 0 & \frac{db}{I_x} \\ \frac{db}{I_y} & 0 & -\frac{db}{I_y} & 0 \\ \frac{k}{I_z} & -\frac{k}{I_z} & \frac{k}{I_z} & -\frac{k}{I_z} \end{bmatrix} \quad (44)$$

For simulation purpose, the constants which are going to be used is set as $g = 9.81 \text{ m/s}^2$, $i_x = 0.0820 \text{ kg.m}^2$, $i_y = 0.0845 \text{ kg.m}^2$, $i_z = 0.1377 \text{ kg.m}^2$, $b = 1.2953 \times 10^{-5} \text{ kg.m}$, $d = 0.165 \text{ m}$, $k = 1.0368 \times 10^{-7} \text{ kg.m}^2$, $m = 4.34 \text{ kg}$.

The output of this quadrotor model is defined by the vector $y = [x \ y \ z \ \psi]^T$, so the matrix C can be written as:

$$C_{(4 \times 12)} = \begin{bmatrix} I_{(3 \times 3)} & O_{(3 \times 9)} \\ O_{(1 \times 3)} & L_{(1 \times 3)} \end{bmatrix} \quad (45)$$

where

$$L_{(4 \times 12)} = [0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0] \quad (46)$$

Finally, the linearization model of this quadcopter is:

$$\dot{x} = A_{12 \times 12}x + B_{12 \times 4}u \quad (47)$$

$$y = C_{12 \times 1}x \quad (48)$$

Remark 2. It can be seen from equation (41) to (45) that most of the components of the linearized state space model are zero components.

Remark 3. Here matrices A , B , and C as sparse matrices and the corresponding state equation is called by the systems with sparseness property.

D. Controllability and Observability

Before designing the controller or the observer, it has been checked the controllability and observability of the system. Define the controllability matrix as:

$$P_c = [B \ AB \ A^2B \ \dots \ A^{n-1}B] \quad (49)$$

where n defines the order of the system. In this case, the matrix can be written as:

$$P_{c(12 \times 48)} = [B \ AB \ A^2B \ \dots \ A^{11}B] \quad (50)$$

$$P_{c(4 \times 12)} = \begin{bmatrix} O_{(8 \times 4)} & O_{(2 \times 4)} & O_{(6 \times 4)} & R_{(2 \times 4)} \\ M_{(4 \times 4)} & M_{(4 \times 4)} & R_{(2 \times 4)} & O_{(10 \times 4)} \\ M_{(4 \times 4)} & O_{(6 \times 4)} & O_{(4 \times 4)} & O_{(8 \times 4)} \end{bmatrix} \quad (51)$$

where

$$R_{(2 \times 4)} = \begin{bmatrix} -\frac{gc_\delta db}{I_y} & \frac{gc_\delta db}{I_x} & \frac{gc_\delta db}{I_y} & -\frac{gc_\delta db}{I_x} \\ -\frac{gs_\delta db}{I_y} & -\frac{gs_\delta db}{I_x} & \frac{gs_\delta db}{I_y} & \frac{gs_\delta db}{I_x} \end{bmatrix} \quad (52)$$

From the above equation is concluded the following proposition.

Proposition 1. The linearized systems with components in equation (41) to (45) are controllable regardless of the value of α , β , γ , and δ .

Proof. The controllability matrix equation (51) has unique configuration of the matrix element. Only prove that the rank of the controllability matrix will always be full rank is needed, i.e., rank of 12, regardless of the value of α , β , γ , and δ . Using standard reduced row escelon form, the systems should be converted into a perfect triangular matrix thus the prove that the system is controllable can be concluded.

By the definition, the matrix observability can be written as:

$$P_o(48 \times 12) = [C \ CA \ CA^2 \ \dots \ CA^{11}]^T \quad (53)$$

$$P_o(48 \times 12) = \begin{bmatrix} \chi_{1(4 \times 12)} \\ \chi_{2(4 \times 12)} \\ \chi_{3(4 \times 12)} \\ \chi_{4(4 \times 12)} \\ O_{(32 \times 12)} \end{bmatrix}^T \quad (54)$$

where

$$\chi_{1(4 \times 12)} = \begin{bmatrix} I_{(3 \times 3)} & O_{(3 \times 9)} \\ O_{(1 \times 3)} & L_{(1 \times 9)} \end{bmatrix} \quad (55)$$

$$\chi_{2(4 \times 12)} = \begin{bmatrix} O_{(4 \times 6)} & I_{(3 \times 3)} & O_{(4 \times 2)} & Z_{(4 \times 1)} \\ O_{(1 \times 3)} & O_{(1 \times 3)} & & \end{bmatrix} \quad (56)$$

$$\chi_{3(4 \times 12)} = \begin{bmatrix} O_{(4 \times 3)} & N_{(2 \times 2)} & O_{(4 \times 7)} \\ O_{(2 \times 2)} & & \end{bmatrix} \quad (57)$$

$$\chi_{4(4 \times 12)} = \begin{bmatrix} O_{(2 \times 9)} & N_{(2 \times 2)} & O_{(4 \times 1)} \\ O_{(2 \times 2)} & & \end{bmatrix} \quad (58)$$

and

$$Z_{(4 \times 1)} = [0 \ 0 \ 0 \ 1]^T \quad (59)$$

then the observability of the systems as summarized in the following preposition can also be concluded.

Preposition 2. The above linearized system is also always observable regardless of the variation of the value of α , β , γ , and δ .

Proof. The proof is very similar with the proof of the first preposition, thus to save space, it is omitted.

Remark 4. What makes this result interesting is the fact that the controllability and observability depend on the equilibrium point in the beginning, which is at $x = \alpha$, $y = \beta$, $z = \gamma$, and $\psi = \delta$. Eventually, the property of matrix A and B depended only on the yaw angle value.

From the above remark, the system can be controlled by more advanced adaptive controller technique such as gain scheduling without concerning the controllability and observability of the system. For gain scheduling, the controller matrix K will depend on the value of yaw angle ψ and will be both controllable and observable for all ψ .

III. CONTROLLER IMPLEMENTATION

Before discussing the gain scheduling that is going to explain in the next section, the novel method for implementing the controller is proposed. It can be observed from the derived model in the previous section that the linearized dynamics of the quadcopter are dominated by chain of integrator. The controller design strategy is based on exploiting this fact by restructuring

the state variable to its Jordan form to emphasize the sparseness of the dynamics [8]. From the restructured states, a pole placement controller is designed. Both of the designed controllers are then re-transformed to the original state space.

The aim of controller design in this article is to formulate a gain matrix K for state-feedback controller, such that the eigen value of a matrix $(A+BK)$ coincides with the desired dynamics pole. The method employed in this article is based on a notion of matrix similarity. A pair of matrix A and B are called similar if there is a similarity transformation P such that:

$$B = PAP^{-1} \quad (60)$$

One of the interesting characteristics of similarity transforms is that the eigen value of the matrices are preserved under the similarity transformation. In this article, the transformed matrix of a matrix A is denoted as A' .

The similarity transform used in this article are the transformation matrix from the original state notation to a new state notation x' composed of:

$$x' = [\epsilon_x \quad x \quad \dot{x} \quad \alpha_1 \phi_r + \alpha_2 \phi_p \quad \alpha_1 \dot{\phi}_r + \alpha_2 \dot{\phi}_p \quad \epsilon_y \quad y \quad \dot{y} \quad \alpha_3 \phi_r + \alpha_4 \phi_p \quad \alpha_3 \dot{\phi}_r + \alpha_4 \dot{\phi}_p \quad \epsilon_z \quad z \quad \dot{z} \quad \epsilon_y \quad \phi_y \quad \dot{\phi}_y]^T \quad (61)$$

The matrix used for this similarity transform can be found on the appendix.

The resulting matrix from this similarity transform is:

$$A' = \begin{bmatrix} H_{5 \times 5} & 0 & 0 & 0 \\ 0 & H_{5 \times 5} & 0 & 0 \\ 0 & 0 & H_{5 \times 5} & 0 \\ 0 & 0 & 0 & H_{5 \times 5} \end{bmatrix} \quad (62)$$

with $H_{n \times n}$ is an $n \times n$ matrix such that:

$$H_{n \times n} = \begin{bmatrix} 0 & I_{n-1 \times n-1} \\ 0 & 0 \end{bmatrix} \quad (63)$$

By changing the last row of H matrix with row vector $[-a_1 \dots -a_n]$ the H matrix become a controller canonical matrix with characteristics polynomial as:

$$H(s) = a_1 + a_2 s^2 + \dots + a_n s^n + s^{n+1} \quad (64)$$

The other interesting property of A' is that by separation principle, the poles of each H can designed without regarding the poles of other blocks. This property permits to do pole placement of each H matrix separately. For a $H_{n \times n}$ matrix, n number of poles ($p_1 \dots p_n$) are selected. The desired characteristics polynomial is given by:

$$H_{id}(s) = \prod_{i=1}^n (s - p_i) = a_1 + a_2 s^2 + \dots + a_n s^n + s^{n+1} \quad (65)$$

The coefficients of the desired polynomial $(-a_1 \dots -a_n)$ are then inserted as $[-a_1 \dots -a_n]$ to the last row of H . Such the desired dynamics matrix of the systems is given by:

$$A'_{id} = A' + \underbrace{(BK)'}_{B'_{id}} \quad (66)$$

The B'_{id} is the collection of H blocks last rows that control the dynamics of the systems. By simple algebraic manipulation, the desired B and K can be derived from a B'_{id} . The K are calculated as:

$$K = B^\dagger (P^{-1} B'_{id} P) \quad (67)$$

The B^\dagger matrix is defined such that for every vector v this relation hold:

$$v = B^\dagger B v \quad (68)$$

Matrix B has a special structure as:

$$B^\dagger = [O_{4 \times 8} \quad B_{sub}^T \quad O_{4 \times 4}] \quad (69)$$

then matrix B^\dagger can be defined as:

$$B^\dagger = [O_{4 \times 8} \quad B_{sub}^{-1} \quad O_{4 \times 4}] \quad (70)$$

The B_{sub} matrix has same structure with the Haar Wavelet Analysis matrix [9], these structures ensures that B_{sub} matrix are invertible as long the coefficients are not zero.

A. Block Diagram of Controller

In this work, the gain will be computed using LQR via gain scheduling. However, pole placement is used in this section for clarity. The naive implementation of the controller will need 48 multiplication and addition. This stems from the fact that the designed gain matrix is not sparse under the original state base. The approach in implementing the designed controller is by using a pre-filter and post-filter before the controller to transform the signal between bases.

The formula of gain as given in equation (67) can be divided as follows.

$$K = \underbrace{B^\dagger P^{-1}}_{\text{Post-filter}} \underbrace{\widehat{B}_{id}}_{\text{Controller}} \underbrace{P}_{\text{Pre-filter}} \quad (71)$$

The value of B^\dagger are fully parameterized by the physical construction of the quadcopter. Thus the gains of this part are static for each quadcopter type. This fact is reflected in designing the controller by only using static gain and adder for this part of post-filter. The P^{-1} part of the post-filter would only permute the position of control signal, thus does not need any mathematical operation to be implemented. The block diagram of the implementation is shown in Figure 2.

The pre-filter part of the implementation is composed of permutation of state structure, eight multiplication blocks and four addition blocks. In the original gain-scheduling method in the next section, the entire gain has to be calculated for

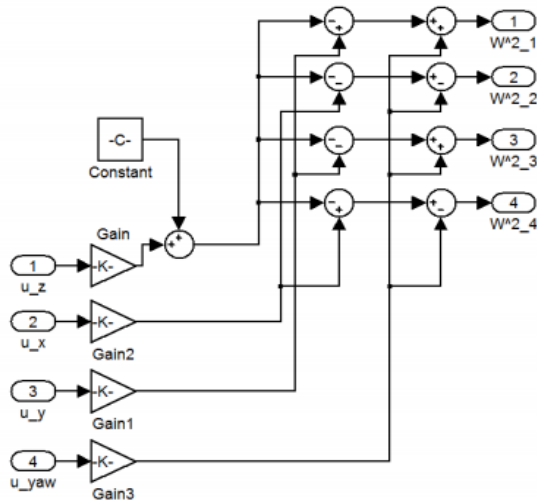


Figure 2. Block Diagram Implementation of post – filter

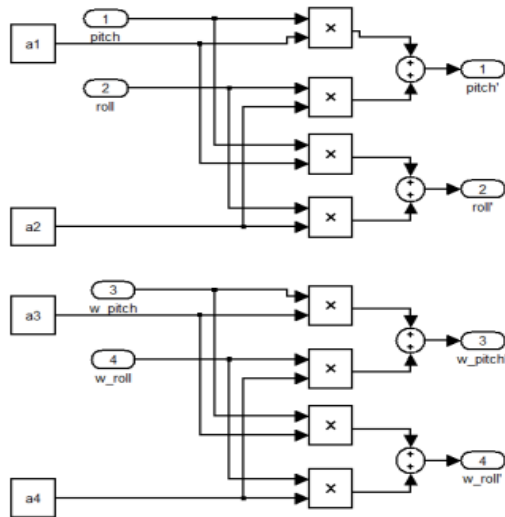


Figure 3. Block Diagram Implementation of pre-filter

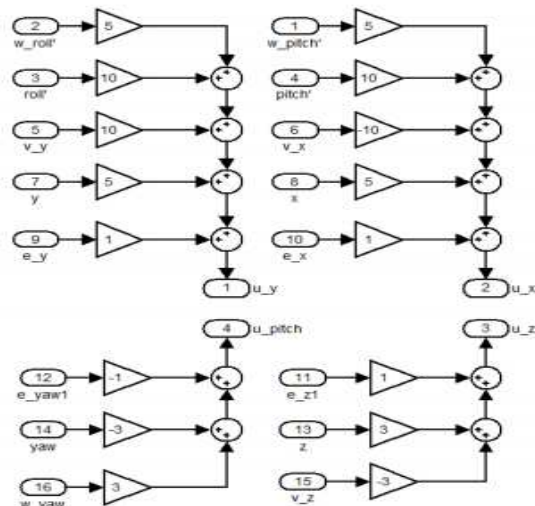


Figure 4. Block Diagram Implementation of controller

each yaw value. However, by using this implementation the equivalent process is achieved by changing the coefficient $\{a_1, a_2, a_3, a_4\}$. The block diagram of the implementation is shown in Figure 3. The implementation for the controller with desired poles in -1 is shown in the block diagram in Figure 4.

B. Numerical Experiment

The numerical experiment is conducted by using Quadrotor Model from Peter Corke's Robotic Toolbox [1]. The controller is implemented using simulink block. There is two numerical experiment performed. The first experiment shows the ability of the controller to achieve desired height. The second experiment shows the ability of the controller to stabilize its height given a force impulse.

Time constant for controller in Figure 5 is 0.998, it closely resembles a linear system with corresponding poles in -1, while the time constant for controller in Figure 6 is 0.644. The controlled system also differs with the linear system by the existence of overshoot in both controllers. The second experiment is conducted by giving force impulse during the period of 7s to 8s. This experiment is aimed to shows the ability of the controller to correct disturbance due to external forces. The experiment is conducted using a controller with poles in -2. The result of this experiment is shown in Figure 7. It is shown

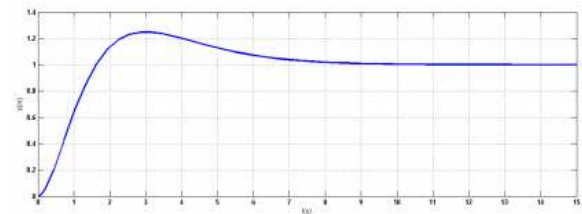


Figure 5. Step Response for controller with -1 poles

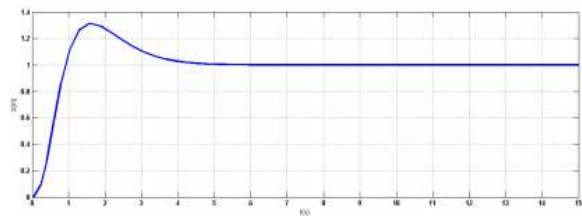


Figure 6. Step Response for controller with -2 poles

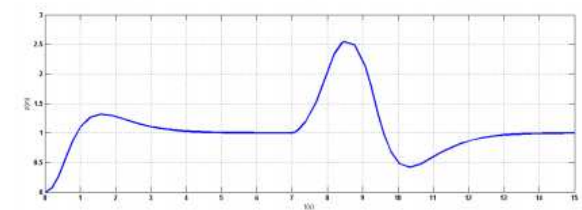


Figure 7. Controller Response to Disturbance

that the controller is able to correct the error due to force impulse.

IV. CONTROL SYSTEM DESIGN

Knowing the controllability and observability property that is independent of the position and yaw movement of the systems, the usage gain scheduling controller is opted. For simplicity, the gain in each of system's structural changing due to the change of eign structure is going to be computed using LQR formalism. Moreover, in order to give more insight, hereby the controller directly was implemented into the original nonlinear system equation (7) to (23) using the series of numerical studies.

A. Linear Quadratic Regulator

Linear quadratic regulator is a control method using state feedback law $u = -Kx$ to minimize the cost function, defined as:

$$J(u) = \int_0^\infty x^T Q x + u^T R u dt \quad (72)$$

where Q is weight matrix for state energy and R is weight matrix for input energy. The matrix K can be derived from equation (73).

$$K = R^{-1}(B^T S + N^T) \quad (73)$$

while the matrix S is solution for the Riccati equation:

$$A^T S + SA - (SB + N)R^{-1}(B^T S + N^T) + Q = 0 \quad (74)$$

In order to solve matrix K , one have to give numerical value for δ so that all numerical values of matrix A and B can be obtained. For simplicity, choose the weight matrix Q and R as $Q = qI_{12 \times 12}$, $R = rI_{4 \times 4}$, where $q = 10,000,000,000$ and $r = 0.0000000001$.

A small value for weight matrix R is chosen because of the minimal energy of input signal is desired. In order to make the quadrotor able to maintain its altitude, a great amount of energy for

input signal is used. However, in order to include the effect of motor saturation, a maximum and minimum boundary of the motor's rotational speed is also used (which is square root of the input signal) according to the [1], where $\omega_{i,max} = 1,000 \text{ rpm}$ and $\omega_{i,min} = 700 \text{ rpm}$ for all $i = \{i : 1 \leq i \leq 4 | i \in \mathbb{Z}\}$.

Here, it was choosen the value for $\delta = 0.5$. The matrix K obtained from the LQR is:

$$K_{(4 \times 12)} = \xi \times [U_{(4 \times 4)} \quad V_{(4 \times 4)} \quad W_{(4 \times 4)}] \quad (75)$$

where ξ is a constant, its value is 1.0×10^{10} and

$$U_{(4 \times 4)} = \begin{bmatrix} -0.6205 & -0.3390 & -0.5000 & 0.0000 \\ 0.3390 & -0.6250 & -0.5000 & -3.8392 \\ 0.6205 & 0.3390 & -0.5000 & 0.0000 \\ -0.3390 & 0.6250 & -0.5000 & 3.8392 \end{bmatrix}, \quad (76)$$

$$V_{(4 \times 4)} = \begin{bmatrix} 3.8392 & 0.5000 & -0.9007 & -0.4921 \\ 0.0000 & -0.5000 & 0.4921 & -0.9007 \\ -3.8392 & 0.5000 & 0.9007 & 0.4921 \\ 0.0000 & -0.5000 & -0.4921 & 0.9007 \end{bmatrix}, \quad (77)$$

$$W_{(4 \times 4)} = \begin{bmatrix} -0.5000 & 0.0000 & 0.7071 & 0.5000 \\ -0.5000 & -0.7071 & 0.0000 & -0.5000 \\ -0.5000 & 0.0000 & -0.7071 & 0.5000 \\ -0.5000 & 0.7071 & 0.0000 & -0.5000 \end{bmatrix} \quad (78)$$

Using the matrix, which has been mentioned above to produce input signal based on the state feedback law $u = -Kx$ and fed the input signal to then on-linear model of quadrotor. In this part, the result for non-linear model quadrotor attitude and altitude control using the state feedback controller that is obtained by using the linearized model property of quadrotor is presented.

Figure 8 shows the quadrotor altitude with initial condition $z = 0$ to steady state condition $z = 5$. While, Figure 9, shows the quadrotor yaw angle with initial condition $\psi = 0$ to steady state condition $\psi = 0.5$. As the picture shown, the state feedback controller obtained from the linearized model of the quadrotor can be concluded to works well. There is no overshoot seen and the system does not need much time to reach the stability. When changing the value of variable $\delta = 1$, the result was got as seen in Figure 10 and

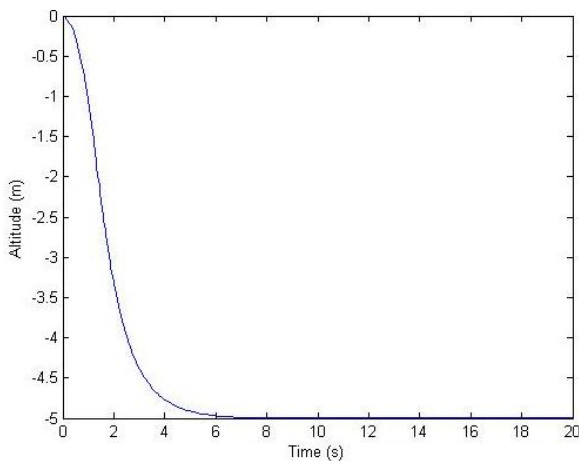


Figure 8. Altitude z , for $\delta = 0.5$

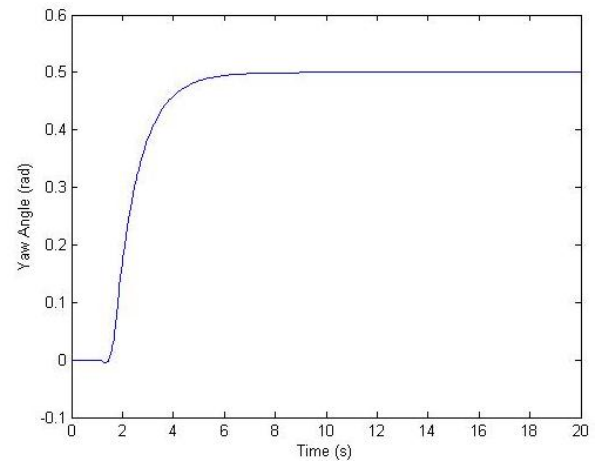


Figure 9. Altitude yaw, for $\delta = 0.5$

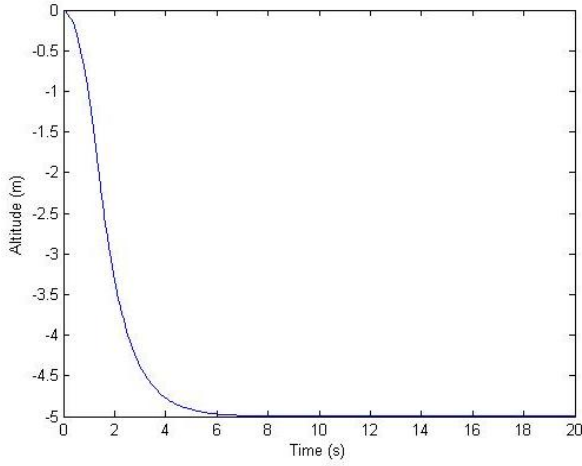
Figure 10. Altitude Z , for $\delta = 1$

Figure 11. Through this simulation, it can be concluded that proposed controller is effective.

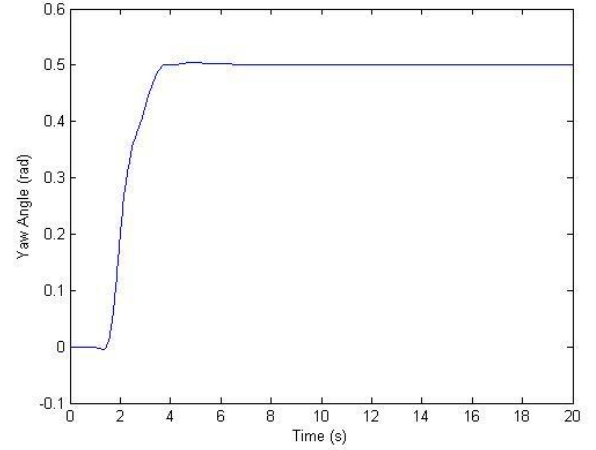
B. Observer

To verify the observability of the system, linear feedback observer also known as Luenberger Observer is applied. The state space equation of the estimated state can be written as follow.

$$\dot{\hat{x}} = f(\hat{x}) + L(y - C\hat{x}) \quad (79)$$

where L is the Luenberger matrix gain and $f(x)$ is nonlinear function describing the quadrotor model as explained in equation (25) until (36).

It can be determined that the matrix L using the pole placement strategy. The eigen value of matrix $(A + LC)$ is designed to have value 1.5 times the poles of the controller, which is the eigen value of matrix $(A + KB)$. To make sure that the error of the observer, the difference between the real state and estimated state, is close to zero fast enough before the states value is

Figure 11. Altitude yaw, for $\delta = 1$

forced to close to the desired value by the controller. The value of matrix L is:

$$L_{(12 \times 4)} = \zeta \times [X_{(4 \times 4)} \quad Y_{(4 \times 4)} \quad Z_{(4 \times 4)}]^T \quad (80)$$

where ζ is a constant, its value is 1.0×10^6 and

$$X_{(4 \times 4)} = \begin{bmatrix} 0.5319 & -0.0475 & 0.0081 & 0.0005 \\ -0.0475 & 0.0438 & 0.0319 & -0.0091 \\ 0.0080 & 0.0319 & 0.1124 & -0.1307 \\ -0.9690 & 0.1999 & 0.0788 & -0.0269 \end{bmatrix}, \quad (81)$$

$$Y_{(4 \times 4)} = \begin{bmatrix} -1.4502 & 0.0675 & -0.0730 & 0.0128 \\ 0.0005 & -0.0091 & -0.1307 & 0.5135 \\ 4.3321 & -0.3868 & 0.0656 & 0.0042 \\ -0.3867 & -0.3568 & 0.2596 & -0.0741 \end{bmatrix}, \quad (82)$$

$$Z_{(4 \times 4)} = \begin{bmatrix} 0.0121 & 0.0478 & 0.1686 & -0.1960 \\ -1.0013 & 0.2066 & 0.0814 & -0.0278 \\ -1.4986 & 0.0698 & -0.0754 & 0.0132 \\ 0.0008 & -0.0136 & -0.1960 & 0.7702 \end{bmatrix} \quad (83)$$

In Figure 12 until 15, the real states drew can be seen in blue line and the estimated states in red line are convergent. For all the states, the observer seems to works well. Through this simulation, the observability of the linearized model of quadrotor has been approved.

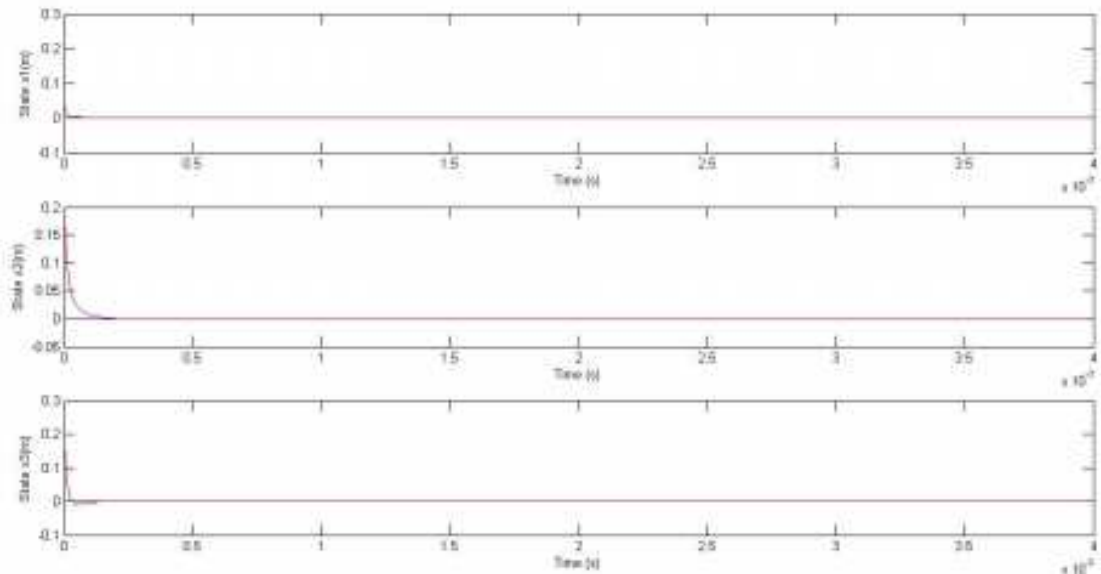


Figure 12. State 1 to state 3

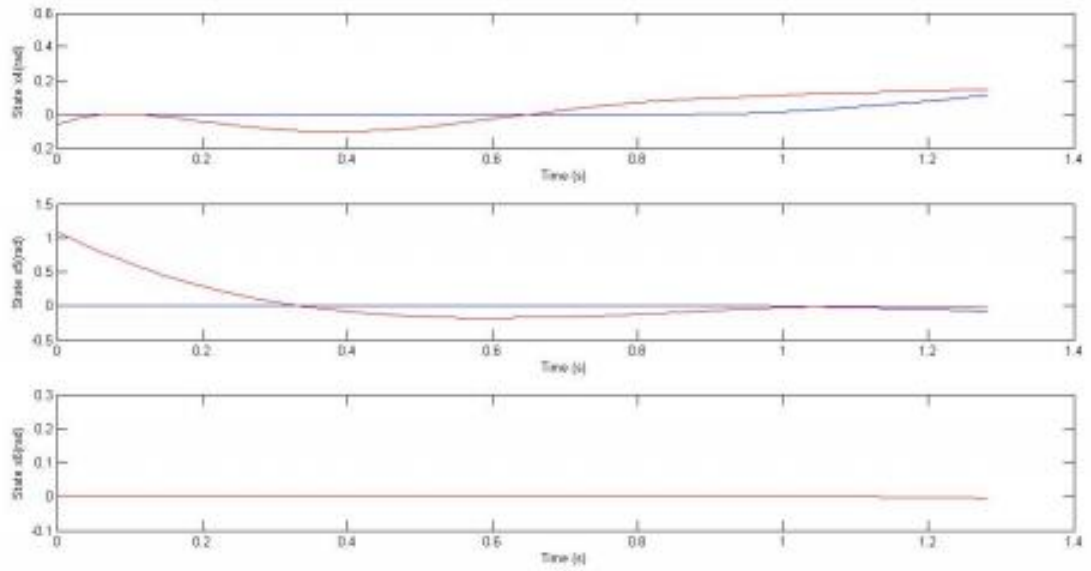


Figure 13. State 4 to state 6

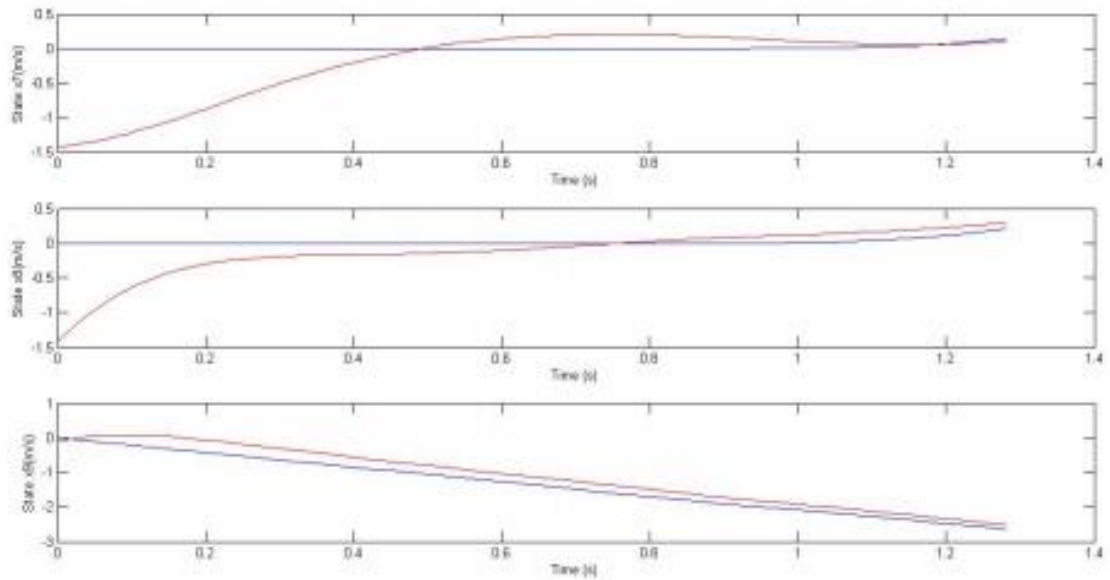


Figure 14. State 7 to state 9

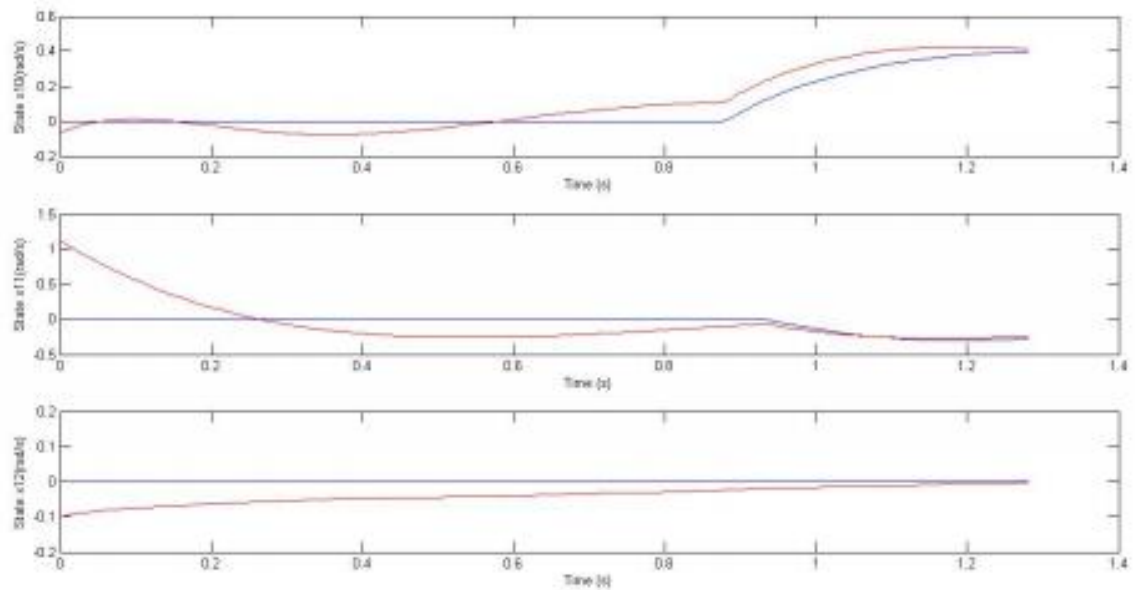


Figure 15. State 10 to state 12

V. CONCLUSION

In this paper, the linearized model of quadrotor simplified model is obtained and show that the system is controllable and observable regardless of the value of reference position in Cartesian coordinate as well as the reference yaw angle. It is found from the linearization that the system has mostly zero components thus can be considered as a sparse system. By rearranging into its respective Jordan form finally via similarity transformation the number of components can be reduced from 48 to 8 plus one permutation and one addition blocks. A numerical study of the implementation said that the scenario worked well. Finally, the gain scheduling controller whose gains are designed via LQR approach is proposed. The simulation results said that the proposed controller that is implemented in the original system using the sparseness property was effective.

APPENDIX

The similarity matrix for transforming the states is given as:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & a_1 & a_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & a_1 & a_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & a_3 & a_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & a_3 & a_4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

ACKNOWLEDGEMENT

The authors would like to thanks to Japan International for Cooperation Agency (JICA) for

their kind support in providing the Collaborative Grant for Researches Alumnae.

REFERENCES

- [1] P. I. Corke, *Robotics, vision and control: fundamental algorithms in MATLAB*, ser. Springer tracts in advanced robotics. Berlin: Springer, 2011, no. v. 73.
- [2] P. Pounds, et al., "Modelling and control of a quad-rotor robot," in *Australasian Conference on Robotics and Automation*, Auckland, New Zealand: Australian Robotics and Automation Association Inc., 2006.
- [3] S. Bouabdallah, et al., "PID vs LQ control techniques applied to an indoor micro quadrotor" in *Intelligent Robots and Systems (IROS 2004)*, vol 3, 2004, pp. 2451-2456.
- [4] S. Gaikwad, et al., "Auto-tuning PID using loopshaping ideas," *Proceedings of the 1999 IEEE International Conference on Control Applications*, vol. 1, 1999, pp. 589-593.
- [5] X. Liu, et al., "Design of self-adaptive PID controller based on least square method." *3rd International Conference on Genetic and Evolutionary Computing, IEEE.*, Oct. 2009, pp. 527-529.
- [6] N. Tamami, et al., "Proportional Derivative Active Force Control for "x" Configuration Quadcopter", *Journal of Mechatronics, Electrical Power, and Vehicular Technology*, Vol.5, pp. 67-74, 2014. DOI: 10.14203/j.mev.2014.v5.67-74. [Online]. Available: www.mevjournal.com
- [7] H. K. Khalil, *Nonlinear systems*, 2nd ed. Upper Saddle River, NJ: Prentice Hall, 1996.
- [8] K. Busawon, "Control design using jordan controllable canonical form," *Proceedings of the 39th IEEE Conference on Decision and Control*, vol. 4, 2000, pp. 3386–3391.
- [9] S. G. Mallat, *A wavelet tour of signal processing the Sparse way*. Amsterdam; Boston: Elsevier /Academic Press, 2009.